CDS

TECHNICAL MEMORANDUM NO. CIT-CDS 93-006
May 6, 1993

"Significance Regression: Robust Regression for Collinear Data"

Tyler R. Holcomb and Manfred Morari

Control and Dynamical Systems
California Institute of Technology
Pasadena, CA 91125

Significance Regression: Robust Regression for Collinear Data

Tyler R. Holcomb

Manfred Morari *

Chemical Engineering 210-41
California Institute of Technology
Pasadena CA 91125

Keywords: significance regression, biased regression, PLS, multivariable regression, robust regression, collinearity

CIT-CDS Technical Memo 93-006

May 6, 1993

Abstract

This paper examines robust linear multivariable regression from collinear data. A brief review of M-estimators discusses the strengths of this approach for tolerating outliers and/or perturbations in the error distributions. The review reveals that M-estimation may be unreliable if the data exhibit collinearity. Next, significance regression (SR) is discussed. SR is a successful method for treating collinearity but is not robust. A new significance regression algorithm for the weighted-least-squares error criterion (SR-WLS) is developed. Using the weights computed via M-estimation with the SR-WLS algorithm yields an effective method that robustly mollifies collinearity problems. Numerical examples illustrate the main points.

*Author to whom correspondence should be addressed: phone (818)356-4186, fax (818)568-8743, e-mail mm@imc.caltech.edu

1 Introduction

This paper examines robust multivariable regression for linear problems of the form

$$y = Xr + e \tag{1}$$

where $X \in \Re^{n_s \times n_i}$ and $y \in \Re^{n_s}$ are the n_s observations of the explanatory and dependent variables respectively and $e \in \Re^{n_s}$ is an unobservable vector of errors. This work focuses on collinear problems, so the main results will be most applicable to problems where the condition number of $X^T X$ is "large." Collinearity will tend to be a problem if there are any correlations among the explanatory variables. Such correlations are common when large numbers of explanatory variables are used, particularly if several explanatory variables are measuring physically similar quantities (*e.g.* using redundant sensors). Problems with multiple dependent variables ($Y \in \Re^{n_s \times n_o}$) can also be treated via "stacking" [Holcomb et al., 1993] to convert vector output regression problems to scalar output regression problems.

The ordinary least squares (OLS) estimate of r is

$$\tilde{r} = (X^T X)^{-1} X^T y. \tag{2}$$

However $\tilde{r}$ has long been known to be an unreliable regressor if the data contain outliers or if the data are collinear. A large selection of methods treat either of these problems, but few systematically and rigorously treat both problems *simultaneously*. Robustness can be achieved by choosing a better objective function than least-squares. Such regressors are called M-estimators and possess both a strong theoretical foundation and a successful history of practical use. The M-regressors can be expressed using a weighted least-squares objective function but can be unreliable for collinear data. A successful rigorous method for treating collinearity, significance regression (SR), is shown to have poor robustness properties. In this paper a robust regressor tolerant of collinearities is developed using M-estimation to generate the weights for the weighted least-squares significance regression method. The resulting significance regressor inherits the robustness properties of the M-estimator while maintaining SR's ability to treat collinearity.

2 Robust Regression

Robust estimation is supported by a rich and successful corpus of theory; only brief portions of the theory needed to develop a robust significance regressor are touched upon here. The interested reader is referred to Huber [Huber, 1964, Huber, 1977] and Tukey [Hoaglin et al., 1983] for further development, as the robustness statements made here are derived from these sources. In this development a limited definition of robustness is used: a regressor is robust if it (1) is insensitive to small deviations of the error distribution from the assumed distribution [Huber, 1977] and (2) remains bounded in the face of small numbers (less than 30%, say) of unbounded gross errors. The second portion of the robustness definition is quantified via the concept of the "breakdown point." As defined by Hampel [Hampel, 1968], the breakdown point of an estimator is the largest possible fraction of the observations for which there is a bound on the change in the estimate when that fraction of the sample is altered without restriction. Thus, a necessary condition for robustness is a non-zero breakdown point. Notably, most of the estimators derived from the (unweighted) least squared error objective, including $\tilde{r}$ of equation 2, have a zero breakdown point: if any given observation is altered without bound, the regressor is also altered without bound.

There are numerous robust estimators to choose from; here the class examined is the M-estimators. M-estimators typically have high breakdown points [Hoaglin et al., 1983] and have been shown to have superior robust regression performance [Huber, 1977]. The M-estimator $\mu_{n_s}(\psi_1, \ldots, \psi_{n_s})$ given the function $\rho(\cdot\,; \mu)$ and the sample $\psi_1, \ldots, \psi_{n_s}$ is the value of μ that minimizes the objective function $\sum_{j=1}^{n_s} \rho(\psi_j; \mu)$ [Goodall, 1983]. A necessary condition M-estimators must satisfy is

$$\frac{\partial \sum_{j=1}^{n_s} \rho(\psi_j; \mu)}{\partial \mu} = 0, \tag{3}$$

which can be expressed as

$$\sum_{j=1}^{n_s} \Psi(\psi_j; \mu) = 0 \tag{4}$$

where $\Psi(\psi_j; \mu) = \frac{\partial \rho(\psi; \mu)}{\partial \mu}\Big|_{\psi=\psi_j}$; typically $\rho(\psi_j; \mu)$ is differentiable with respect to μ almost everywhere.

The two most familiar M-estimators are the sample mean, derived from $\rho_1(\psi_j; \mu) = \frac{1}{2}(\psi_j - \mu)^2$, and the sample median, derived from $\rho_2(\psi_j; \mu) = |\psi_j - \mu|$. For reasons

of computational efficacy or analytical tractability one often desires to describe $\rho(\psi_j;\mu)$ using the weighted least squares squares objective, $\rho_3(\psi_j;\mu) = (\omega_j/2)(\psi_j - \mu)^2$. For any given $\rho(\psi_j;\mu)$, $\rho_3(\psi_j;\mu)$ will yield the same μ as the solution of equation 4 if $\omega_j = \Psi(\psi_j;\mu)/(\mu - \psi_j)$. To develop robust regressors, an additional constraint will be added to the functional form of ρ: $\rho(\psi_j;\mu) = \rho(\Delta_j)$ where $\Delta_j = \psi_j - \mu$. Also, for regression, one must be able to define what is "large" and what is "small" for the sake of identifying outliers. This is done via the scale parameter σ_{robust}, which itself is a robust dispersion estimator. Defining the j-th observation of the dependent variable y to be ψ_j, the robust regressor $\tilde{r}_{\text{robust}}$ and scale parameter σ_{robust} are defined by the minimization problem

$$\{\tilde{r}_{\text{robust}}, \sigma_{\text{robust}}\} = \arg \min_{v \in \Re^{n_i},\ \sigma > 0} \sum_{j=1}^{n_s} \sigma \rho\left(\frac{\psi_j - v^T x_j}{\sigma}\right) + \alpha\sigma \tag{5}$$

where $\alpha \in \Re$ is specified so that $\tilde{r}_{\text{robust}}$ is an unbiased estimator of r. To compute α, let γ be a random variable with the same distribution as the elements of e; then $\alpha = (n_s - n_i)\mathcal{E}(\xi(\gamma))$. The function $\xi(\cdot)$ is defined following equation 6. The objective function in equation 5 is convex, so either an infimum occurs at the boundary $\sigma_{\text{robust}} = 0$ or the minimum is specified by the $n_i + 1$ equations

$$\sum_{j=1}^{n_s} \Psi\left(\frac{\Delta_j}{\sigma_{\text{robust}}}\right) x_j = 0 \quad \text{and} \quad \sum_{j=1}^{n_s} \xi\left(\frac{\Delta_j}{\sigma_{\text{robust}}}\right) = \alpha \tag{6}$$

where $\xi(\Delta_j/\sigma_{\text{robust}}) = [\Delta_j \Psi(\Delta_j/\sigma_{\text{robust}})]/\sigma_{\text{robust}} - \rho(\Delta_j/\sigma_{\text{robust}})$ and $\Delta_j = \psi_j - x_j^T \tilde{r}_{\text{robust}}$. One should note that in equation 6 $\Psi(\psi) = \partial\rho(\psi)/\partial\psi$ for any scalar ψ.

As discussed above, any given M-estimation objective ρ can be re-expressed using the weighted least squares objective ρ_3. Likewise, the dual M-estimation problem posed in equations 6 is equivalent to minimizing the weighted least squares objective

$$\arg \min_{v \in \Re^{n_i}} (y - Xv)^T M_{\text{robust}} (y - Xv) \tag{7}$$

for the diagonal matrix $M \in \Re^{n_s \times n_s}$ whose diagonal elements are the ω_j used for $\rho_3(\Delta_j/\sigma_{\text{robust}})$; Huber [1977] discusses this equivalence in greater detail. Solving the equations 6 leads to the following iteratively re-weighted least squares (IRLS) algorithm:

Algorithm 1 (Robust M-Regression)

$$i = \quad 0 \tag{8}$$

$$\tilde{r}_0 = \quad (X^T X)^{-1} X^T y \tag{9}$$

$$\sigma_0 = \frac{\|X\tilde{r}_0 - y\|}{\sqrt{n_s - n_i}} \tag{10}$$

$$\Delta_j = \psi_j - x_j^T \tilde{r}_0 \quad \forall j \tag{11}$$

$$M_0 = \text{Diag}\left(\frac{\Psi(\Delta_j/\sigma_0)}{\Delta_j/\sigma_0}\right) \tag{12}$$

DO

$$i = i + 1 \tag{13}$$

$$\tilde{r}_i = (X^T M_{i-1} X)^{-1} X^T M_{i-1} y \tag{14}$$

$$\Delta_j = \psi_j - x_j^T \tilde{r}_i \quad \forall j \tag{15}$$

$$\sigma_i = \sigma_{i-1}\sqrt{\frac{1}{\alpha}\sum_{j=1}^{n_s} \xi\left(\frac{\Delta_j}{\sigma_{i-1}}\right)} \tag{16}$$

$$M_i = \text{Diag}\left(\frac{\Psi(\Delta_j/\sigma_i)}{\Delta_j/\sigma_i}\right) \tag{17}$$

UNTIL convergence

$$M_{\text{robust}} = M_i \tag{18}$$

$$\sigma_{\text{robust}} = \sigma_i \tag{19}$$

$$\tilde{r}_{\text{robust}} = \tilde{r}_i. \tag{20}$$

The scalar α in equation 16 was defined following equation 5. This algorithm is adapted from Huber's **Algorithm H** [Huber, 1977] and has been proven to converge uniquely. In fact, this algorithm converges quickly, typically in less than ten iterations.

An open question is the choice of $\rho(\cdot)$. A common and successful [Chow, 1983] M-estimator uses Huber's "proposition 2" objective function:

$$\rho\left(\frac{\Delta_j}{\sigma_{\text{robust}}}\right) = \begin{cases} \frac{\Delta_j^2}{2\sigma_{\text{robust}}^2} & \text{for } |\frac{\Delta_j}{\sigma_{\text{robust}}}| \leq 1 \\ |\frac{\Delta_j}{\sigma_{\text{robust}}}| - \frac{1}{2} & \text{for } |\frac{\Delta_j}{\sigma_{\text{robust}}}| > 1 \end{cases} . \tag{21}$$

For this $\rho(\cdot)$, $\alpha = 0.258$. One can see that when using Huber's "proposition 2" objective the resulting M-estimator uses the mean, which is "efficient" (in the classical sense) but not robust for "small" errors, and uses the median, which is robust but not "efficient," for "large" errors .

For particularly heavy-tailed error distributions (numerous outliers), a better $\rho(\cdot)$ may

be Tukey's biweight [Hoaglin et al., 1983]:

$$\rho\left(\frac{\Delta_j}{\sigma_{\text{robust}}}\right) = \begin{cases} \frac{1}{6}\left(1-\left(1-\frac{\Delta_j^2}{\sigma^2}\right)^3\right) & \text{for } |\frac{\Delta_j}{\sigma_{\text{robust}}}| \leq 1 \\ \frac{1}{6} & \text{for } |\frac{\Delta_j}{\sigma_{\text{robust}}}| > 1 \end{cases} . \tag{22}$$

One should keep in mind that the theory supporting M-regressors assumes independent and *a priori* homoscedastic errors; therefore one should always scale the data such that $\mathcal{E}\left(ee^T\right) = \sigma_e^2 I$ before using algorithm 1. More importantly, M-estimators in general, and these two in particular, are *not* the minimum-variance unbiased estimators; achieving the minimum-variance property comes at the direct expense of robustness. However, both the "proposition 2" and Tukey's biweight objective functions lead to unbiased estimators with breakdown points of almost 0.5 [Goodall, 1983].

While algorithm 1 produces robust regressors, it inherits OLS's weakness for collinear problems. This weakness is easily seen in the variance of the robust regressor. To compute this variance one can make the simplifying assumption that M_{robust} is independent of e. Then

$$\text{Var}(\tilde{r}_{\text{robust}}) = (X^T M_{\text{robust}} X)^{-1} X^T M_{\text{robust}} P M_{\text{robust}} X (X^T M_{\text{robust}} X)^{-1} \sigma_e^2. \tag{23}$$

Thus one can see that the estimate will tend to be unreliable when $X^T M_{\text{robust}} X$ is ill-conditioned (collinear).

3 Significance Regression for the Classical Model

To develop an understanding of how to treat the collinearity problem that bedevils $\tilde{r}_{\text{robust}}$, this section reviews an approach to collinearity for the classical model (equation 1). One can mollify the collinearity problem for classical regression by employing techniques such as stepwise regression [Draper and Smith, 1966], ridge regression [Hoerl and Kennard, 1970], principal components regression [Hill et al., 1977], and significance regression (SR) [Holcomb et al., 1993]. The SR approach encompasses the successful partial least squares algorithm [Wold et al., 1984], has been claimed to have better prediction properties than ridge regression [Fearn, 1983] (although this claim is contradicted by the results of Friedman [Frank and Friedman, 1992]), has been shown to have better prediction properties than principal components regression [Lorber et al., 1987, Stone and Brooks, 1990] for a

variety of problems, and rests on a rigorous foundation that can be readily generalized. A comprehensive motivation and derivation for significance regression for the classical model is presented in [Holcomb et al., 1993]; only the main points are considered here.

Typically, the specification of a regressor can be expressed as an unconstrained optimization problem. For example, equation 2 results directly from the minimization problem

$$\tilde{r} = \arg \min_{v \in \Re^{n_i}} (y - Xv)^T (y - Xv). \tag{24}$$

The variance of the regressor can be reduced if one constrains the allowable values for the final regressor. For example ridge regression uses a "soft" constraint derived from assuming a prior distribution for r [Gruber, 1990]. For ridge regression, the appropriate optimization problem is

$$\arg \min_{v \in \Re^{n_i}} (y - Xv)^T (y - Xv) + v^T A v. \tag{25}$$

for some positive definite A that describes the inverse of covariance matrix of the prior distribution (assuming that the expectation of the prior distribution is the origin). Another approach is to constrain the regressor to a prespecified subspace, as in

$$\tilde{b} = \arg \min_{v \in \text{Range}(W)} (y - Xv)^T (y - Xv) \tag{26}$$

where $W \in \Re^{n_i \times n_d}$ consists of orthonormal columns. For stepwise regression each column would consist of unit vectors describing coordinate axes. For example if one chooses to use the second and third of three variables, then $W = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$. For principal components regression W would be built from the principal components of X. But how to choose the "best" W? First, one clearly desires $r \in \text{Range}(W)$ to assure that the regressor is an unbiased estimator. Moreover, if $< w, r > = w^T r = 0$, then w should not be used as a column for W since this will increase the variance without affecting the bias. One can quantify and exploit these observations by postulating the null hypothesis

$$\mathcal{H}_0^1 : \qquad < w, r > = 0 \tag{27}$$

and searching for directions (w) that reject it. A natural test statistic for $\mathcal{H}_0^1$ is

$$\tau_{classical}\ (w, y) = \frac{< w, \tilde{r} >^{\ 2}}{\text{Var}(< w, \tilde{r} >)} \tag{28}.$$

for which $\text{Var}(< w, \tilde{r} >) = \sigma_e^2 w^T (X^T X)^{-1} w$. For any given w and normally distributed errors, $\tau_{classical}\ (w, y)$ has a non-central χ^2 distribution with one degree of freedom; when $\mathcal{H}_0^1$ holds, the non-centrality parameter is zero. Thus, one can build W by seeking mutually orthogonal directions that successively maximize $\tau_{classical}\ (w, y)$; this method is precisely significance regression. For the classical model, the SR algorithm is:

Algorithm 2 (SR)

$$\tilde{r} = \quad (X^T X)^{-1} X^T y \tag{29}$$

$$V = \quad (X^T X)^{-1} \tag{30}$$

$$W_0 = \quad [0 \cdots 0]^T, \quad W_0 \in \Re^{n_i} \tag{31}$$

$$\text{DO } i = \quad 1, n_d$$

$$v = \quad (I - W_{i-1} W_{i-1}^T) V^{-i} \tilde{r} \tag{32}$$

$$w_i^{opt}(y) = \quad \frac{v}{||v||} \tag{33}$$

$$W_i = \quad [w_1^{opt} | w_2^{opt} | \cdots | w_i^{opt}] \tag{34}$$

END DO.

$$\tilde{b} = \quad W_{n_d} (W_{n_d}^T X^T X W_{n_d})^{-1} W_{n_d}^T X^T y \tag{35}$$

To determine n_d, one can use use either cross-validation [Stone, 1974] on the prediction error or one can use hypothesis testing. In particular, if the null hypothesis

$$\mathcal{H}_0^{2,i}: \qquad < w, r > = 0 \quad \text{for all } w \in \text{Range}(I - W_{i-1} W_{i-1}^T) \tag{36}$$

is true then $n_d < i$. One can test $\mathcal{H}_0^{2,i}$ using the test statistic $\tau_{classical}(w_i^{opt}(y), y)$. Let $n_p = n_i - i + 1$. For normal errors and large n_s, the test "if $\tau_{classical}(w_i^{opt}, y) > n_p$ then $n_d \geq i$" is equivalent to using $\tau_{classical}(w_i^{opt}, y)$ to test $\mathcal{H}_0^{2,i}$ at the 50% significance level. Derivation, analysis, and discussion of this significance testing approach are given in [Holcomb et al., 1993].

4 Robust Significance Regression

As shown in the prior section, SR is a systematic approach for mitigating collinearity problems. However any regressor based on equation 25 or equation 26 will have a zero breakdown point and thus will not be robust. On the other had, the weights generated by

algorithm 1 (M_{robust}) insure that regressors produced using equation 7 are robust. Using the SR method to treat collinearity one would compute the regressor from constrained minimization, as in

$$\arg \min_{v \in \text{Range}(W)} (y - Xv) M_{\text{robust}} (y - Xv). \tag{37}$$

Since $\tilde{r}_{\text{robust}}$ is unbiased and its variance can be approximated (equation 23), one can use SR to compute an appropriate W.

Using the development in the previous section, the null hypothesis of interest is $< w, r_{\text{robust}} > = 0$. The natural test statistic is

$$\tau_{\text{robust}}(w, y) = \frac{(w^T \tilde{r}_{\text{robust}})^2}{\text{Var}(w^T \tilde{r}_{\text{robust}})}. \tag{38}$$

Using the simplifying assumption that M_{robust} is independent of e, the variance is

$$\begin{aligned} \text{Var}(w^T \tilde{r}_{\text{robust}}) &= w^T \mathcal{E}\left((r_{\text{robust}} - r)(r_{\text{robust}} - r)^T \right) w & (39) \\ &= w^T \text{Var}(\tilde{r}_{\text{robust}}) w & (40) \end{aligned}$$

leads to

$$\tau_{\text{robust}}(w, y) = \frac{(w^T \tilde{r}_{\text{robust}})^2}{w^T \text{Var}(\tilde{r}_{\text{robust}}) w}. \tag{41}$$

Using this test statistic and robust methods in section 2, a robust significance regression method is:

Algorithm 3 (SR - Robust)

Rescale		the data such that $P = \sigma_e^2 I$	(42)
Choose		an $M-$estimation objective function, $\rho(\cdot)$	(43)
Compute		M_{robust}, $\tilde{r}_{\text{robust}}$, and σ_{robust} with algorithm 1	(44)
$V =$		$(X^T M_{\text{robust}} X)^{-1} X^T M_{\text{robust}}^2 X (X^T M_{\text{robust}} X)^{-1}$	(45)
$W_0 =$		$0, \quad W_0 \in \Re^{n_*}$	(46)
DO $i =$		$1, n_d$	
	$v =$	$(I - W_{i-1} W_{i-1}^T) V^{-i} \tilde{r}_{\text{robust}}$	(47)
	$w_i^{opt} =$	$\frac{v}{\|\|v\|\|}$	(48)
	$W_i =$	$[w_1^{opt} \| w_2^{opt} \| \cdots \| w_i^{opt}]$	(49)

END DO

$$\tilde{b}_{\text{robust}} = W_{n_d}(W_{n_d}^T X^T M_{\text{robust}} X W_{n_d})^{-1} W_{n_d}^T X^T M_{\text{robust}} y. \tag{50}$$

Notice that the resulting regressor,

$$\tilde{b}_{\text{robust}} = \arg \min_{v \in \text{Range}(W_{n_d})} (Xv - y)^T M_{\text{robust}} (Xv - y), \tag{51}$$

has the same breakdown point as $\tilde{r}_{\text{robust}}$ and inherits *all* of the robustness properties of the M-estimator when $r \in \text{Range}(W_{n_d})$. Thus using significance regression does not cause "loss of robustness" but does maintain the ability to treat collinearities.

Algorithm 3 does not firmly specify how to choose n_d. A reasonable approach is to perform cross-validation using the robust objective function in equation 51. Alternatively one can develop a useful test from the significance regression framework. Following the approach described in section 3, the "robust" test statistic of equation 41 can be computed using the V of equation 45

$$\tau_{\text{robust}}(w, y) = \frac{(w^T \tilde{r}_{\text{robust}})^2}{w^T V^{-1} w \sigma^2_{\text{robust}}}. \tag{52}$$

and employed in the decision rule "if $\tau_{\text{robust}}(w_i^{opt}, y) > n_p$ then $n_d \geq i$." This decision rule does not rest on theoretical derivation; in particular, it is not equivalent to the 50% significance test for large n_s. However, as will be demonstrated next section, it does provide a useful *ad hoc* rule.

Algorithm 3 is not the only method for combining M-estimation and significance regression. Re-expressing equation 26 as

$$\tilde{b} = \arg \min_{v \in \text{Range}(W)} \sum_{j=1}^{n_s} \rho_1(\psi_j; v^T x_j), \tag{53}$$

one can see that another reasonable approach is to compute

$$\tilde{b} = \arg \min_{v \in \text{Range}(W)} \sum_{j=1}^{n_s} \rho(\psi_j; v^T x_j) \tag{54}$$

where $\rho(\cdot, \cdot)$ is an appropriate M-estimation objective function. Algorithm 3 does not produce the solution to equation 54 since Δ_j's produced by the solution to equation 54 will be different from the Δ_j's produced using $\tilde{b}_{\text{robust}}$. A conjectured solution to equation 54 is to iterate algorithm 3. In such an iterative algorithm, the first iteration would be precisely

algorithm 3, and all further iteration would use algorithm 3 with the modification that the $\tilde{r}_{\text{robust}}$ in the current iteration (equations 9 and 14 of algorithm 1) would be constrained to lie in the range of the W computed in the previous iteration. This approach involves nested iterations (for each major iteration, M_{robust} must be recomputed iteratively), so it can be computationally burdensome. Moreover, the benefit of the iterative method is uncertain. The errors (e) possess n_s degrees of freedom (can affect y anywhere in $\Re^{n_o}$), while the corrections to SR-robust produced by iteration possess only n_p degrees of freedom (can affect y only in Range $\left(X(I - W_{n_d} W_{n_d}^T)\right)$). Thus, in this work the algorithm 3 is preferred; further analysis of equation 54 awaits future inquiry.

5 Simulation Examples

This section presents a comparison between the multivariable regression methods discussed in this paper. In this study, the examples are simulation studies using purely synthetic data. The data are not claimed to correspond to any particular "real world" process; rather, the data were generated to conform to the model assumptions and to illustrate the relative effectiveness of various methods for problems that satisfy the model assumptions. The "real world" successes of PLS [Martens and Næs, 1989, Mejdell, 1990, Ricker, 1988] are suggested as evidence of the practical utility of SR since PLS is closely related to SR. The regression methods investigated were

- ordinary least squares (OLS, equation 2),
- M-estimation (M-est, algorithm 1),
- significance regression (SR, algorithm 2), and
- robust significance regression (SR-robust, algorithm 3).

Both robust algorithms used Huber's "proposition 2" objective function; both cross-validation and significance testing were used to choose n_d.

All examples had ten explanatory variables and four dependent variables ($n_i = 10$ and $n_o = 4$); thus, the data conformed to the model

$$Y = XR + E \tag{55}$$

where $Y \in \Re^{n_s \times n_o}$, $X \in \Re^{n_s \times n_i}$, $R \in \Re^{n_i \times n_o}$, and $E \in \Re^{n_s \times n_i}$. One thousand distinct examples were examined to mitigate sampling effects in the numerical results. Each example was generated by the method presented in appendix B. Since both the variances of the explanatory variables and the values of the regression parameters varied over five orders of magnitude and since there were typically large variances among the explanatory variables that had little effect on the dependent variables, this exploration shed light on the relative strengths and weaknesses of the methods for a class of problems that has historically bedeviled OLS.

The examples for this simulation study could easily have been designed to provide SR-robust with an overwhelming advantage over SR since the standard SR algorithm has a zero breakdown point. While this would provide some dramatic numbers, little insight would be gained. Instead this study used data that is only mildly corrupted with outliers: for each error, there was a 5% probability that the error would be drawn form a distribution with three times the standard deviation. According to Huber, "typical 'good data' samples in the physical sciences appear to be well modeled" by this distribution [Huber, 1977, p. 2].

Four measures were employed to evaluate regressor performance. Since the examples were synthetic, R was known and the estimation error could be computed for each example. The measure was

$$RMS_{\text{MSE}} = \sqrt{\frac{\text{Tr}(\ (\tilde{B} - R)(\tilde{B} - R)^T\ )}{\text{Tr}(RR^T)}}. \tag{56}$$

The $\text{Tr}(RR^T)$ term was included to produce a relative error and allow averaging over all one thousand examples. The second measure was computed based on the PRESS. For each example an additional one hundred samples (X_{new}, Y_{new}) were generated from the identical distribution as the training data, but the Y_{new} were not corrupted by error ($E_{new} = 0$). Then

$$RMS_{\text{PRESS}} = \sqrt{\frac{\text{Tr}(\ (X_{new}\tilde{B} - Y_{new})^T(X_{new}\tilde{B} - Y_{new})\ )}{400}}. \tag{57}$$

Since the data were generated with the constraint

$$\sqrt{\frac{\text{Tr}(Y_{new}^T Y_{new})}{400}} = 1 \tag{58}$$

the RMS_{PRESS} was averaged over the examples without normalization. Note that $n_s \times n_o = 400$ for the test set.

These two measures where then averaged over all one thousand examples to produce the mean RMS_{PRESS}, $MEAN_{\text{PRESS}}$, and mean RMS_{MSE}, $MEAN_{\text{MSE}}$. Since the above two measures are averages, these measures themselves are prone to be unduly influenced by outliers (exceptional examples). The $MEAN_{\text{MSE}}$ is particularly vulnerable since the quantity it averages, RMS_{MSE}, involves division by the potentially small number $\text{Tr}(RR^T)$. To develop a less outlier-sensitive measure, for each example the rank (relative performance) of each estimator was recorded: rank = 1 if no other regressor did better for that example, rank = 2 if one other regressor did better, and rank = 3 if two other regressors did better. If two regressors were within 0.1% of each other, they were given the same rank. The average rank with respect to both MSE and PRESS was computed; this average rank will not be unduly influenced by a few extreme examples.

Notice that each sample contains $n_s n_o = 120$ samplings of the error distribution, of which typically only 4 will lie outside the 3σ range of the nominal distribution. While the outliers in this study were generated by perturbing the output, these perturbations are also loosely equivalent to failures of explanatory variable measurements since the errant input value will tend to lead to misprediction and hence appear as an outlier.

To illustrate the nature of the examples, the first simulation compared OLS to the M-estimator; these results are shown in Table 1. OLS actually outperformed the M-estimator: the dominant feature of these examples is collinearity, not outliers. Examining Table 2 reveals the efficacy of the robust method and the "robust" significance test. SR, which (non-robustly) treats the collinearity, brings the $MEAN_{\text{PRESS}}$ down to 0.28, while SR-robust, which addresses the (mild) outliers, has a $MEAN_{\text{PRESS}}$ of 0.22. The robust method was a consistently better regressor as revealed by the ranks.

As discussed in Holcomb *et al.* [Holcomb et al., 1993], the significance test is useful for estimation, but cross-validation is often a better method for choosing n_d for prediction purposes. For SR-robust, cross-validation was performed on the "robust" PRESS: the contribution of each output was weighted by it's M-estimation weight, thereby diluting the effect of outliers. The results are shown in Table 3. For all examples, thirty samples were available for training ($n_s = 30$). Ten-way (three-out) cross-validation was used to determine n_d. The prediction performance improved relative to the results given in Table 2, but the estimation performance of SR suffered; the same examples were used in the two different simulations, so the numbers are directly comparable. Cross-validation does

method	$\overline{\text{rank}_{\text{PRESS}}}$	$MEAN_{\text{PRESS}}$	$\overline{\text{rank}_{\text{MSE}}}$	$MEAN_{\text{MSE}}$
OLS	1.4	0.42	1.4	1000
M – est	1.6	0.43	1.6	1000

Table 1: Comparison of OLS and M-est over 1,000 examples of synthetic data.

method	$\overline{\text{rank}_{\text{PRESS}}}$	$MEAN_{\text{PRESS}}$	$\overline{\text{rank}_{\text{MSE}}}$	$MEAN_{\text{MSE}}$
OLS	2.8	0.42	3.0	1000
SR	1.7	0.28	1.5	1.0
SR – robust	1.4	0.22	1.4	1.4

Table 2: Comparison of SR-robust and SR over 1,000 examples of synthetic data with outliers when using significance testing to choose n_d.

method	$\overline{\text{rank}_{\text{PRESS}}}$	$MEAN_{\text{PRESS}}$	$\overline{\text{rank}_{\text{MSE}}}$	$MEAN_{\text{MSE}}$
OLS	3.6	0.42	3.0	1000
SR	1.6	0.22	1.5	1.3
SR – robust	1.4	0.21	1.3	1.3

Table 3: Comparison of SR and SR-robust over 1,000 examples of synthetic data with outliers when using cross-validation to choose n_d. SR-robust used "robust" PRESS.

method	$\overline{\text{rank}_{\text{PRESS}}}$	$MEAN_{\text{PRESS}}$	$\overline{\text{rank}_{\text{MSE}}}$	$MEAN_{\text{MSE}}$
OLS	2.8	0.36	3.0	870
SR	1.6	0.25	1.5	1.0
SR – robust	1.6	0.22	1.5	1.3

Table 4: Comparison of SR-robust and SR over 1,000 examples of outlier-free synthetic data when using significance testing to choose n_d.

method	$\overline{\text{rank}_{\text{PRESS}}}$	$MEAN_{\text{PRESS}}$	$\overline{\text{rank}_{\text{MSE}}}$	$MEAN_{\text{MSE}}$
OLS	3.0	2.2	3.0	5.1×10^3
SR	2.0	0.87	1.9	8.2
SR – robust	1.0	0.21	1.1	0.9

Table 5: Comparison of SR-robust and SR over 1,000 examples of synthetic data with "gross" outliers when using significance testing to choose n_d.

provide the SR method with a small measure of robustness: if the effect of outliers can be mitigated by reducing n_d, cross-validation will tend to so. This robustness effect may be why the cross-validated SR had the same $MEAN_{\text{PRESS}}$ as SR-robust using the significance test. However, the "robust" cross-validation gave the best prediction performance, as measured by both the $MEAN_{\text{PRESS}}$ and the ranks. However, cross-validation also required ten times more computational effort than significance testing.

The versatility of SR-robust was investigated by altering the error distribution. OLS, SR, and SR-robust were compared over the same one thousand examples but without any outliers: all errors were drawn from the same normal distribution. For these "well-behaved" errors, SR was derived from the minimum-variance unbiased estimator, while SR-robust was not, so one would expect SR to have the advantage. Table 4 shows this to be the case for the most direct measurement of estimation performance, the $MEAN_{\text{MSE}}$, although the difference are not great. Interestingly, comparison to Table 2 shows that the performance of SR-robust is similar on data with and without outliers. This observation casts further doubt on the potential benefit of iterating algorithm 3 to "improve" SR-robust.

The utility of the the robust method for "gross" outliers was revealed by increasing the standard deviation of the outliers by a factor of 10 (from 3 times the nominal to 30 times the nominal). The results are shown in Table 5. Comparing these results to the results in Table 2, one can see that OLS and SR were drastically affected by "gross" outliers, while SR-robust was little affected.

6 Summary

This paper developed a novel robust restriction regressor, SR-robust. This regressor employed the objective functions that make the M-estimators tolerant of outliers and distributionally robust while using the significance regression (SR) method to treat collinearities. By choosing among the well-analyzed M-estimation objective functions, one can "tune" the method if one knows that the error distribution is "heavy-tailed" or if one has other special knowledge of the error distribution. The method proceeds by using an iteratively-reweighted least squares method to generate weights which are then used in a significance regression algorithm for the weighted least squares objective function.

The effectiveness of the SR-robust method for treating collinear data with outliers was illustrated via simulation. In these simulations, SR-robust was seen to provide better estimation and prediction than SR, M-estimation, and ordinary least squares (OLS) for data with "gross" outliers and data with "mild" outliers. Moreover, SR-robust provided comparable performance to SR on outlier-free data. The close kinship of the SR-robust algorithm to partial least squares indicates that the method is practically useful.

Acknowledgments *Partial support this research through the Department of Energy, Office of Basic Energy Sciences is gratefully acknowledged.*

References

Chow, G. C. (1983). *Econometrics.* McGraw-Hill.

Draper, N. R. and Smith, H. (1966). *Applied Regression Analysis.* Wiley.

Fearn, T. (1983). A misuse of ridge regression in the calibration of a near infrared reflectance instrument. *Applied Statistics*, 32(1):73–79.

Frank, I. E. and Friedman, J. H. (1992). A statistical review of some chemometrics regression tools. Technical report, Dept. of Statistics, Stanford University, Stanford, CA 94305.

Goodall, C. (1983). *Understanding Robust and Exploratory Data Analysis*, chapter M-Estimators of Location: An Outline of the Theory. Wiley.

Gruber, M. H. (1990). *Regression Estimators.* Academic Press.

Hampel, F. R. (1968). *Contributions to the theory of robust estimation.* PhD thesis, University of California, Berkeley.

Hill, R. C., Fomby, T. B., and Johnson, S. (1977). Component selection norms for principal components regression. *Communications in Statistics A: Theory and Methods*, A6(4):309–334.

Hoaglin, D. C., Mosteller, F., and Tukey, J. W. (1983). *Understanding Robust and Exploratory Data Analysis.* Wiley.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67.

Holcomb, T. R., Hjalmarsson, H., and Morari, M. (1993). Significance regression: A statistical approach to biased regression and partial least squares. CDS Technical Memo CIT-CDS 93-002, California Institute of Technology, Pasadena, CA 91125.

Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Statistics*, 35:73–101.

Huber, P. J. (1977). *Robust Statistical Procedures.* SIAM.

Lorber, A., Wangen, L., and Kowalski, B. (1987). A theoretical foundation for the pls algorithm. *Journal of Chemometrics*, 5:19–31.

Martens, H. and Næs, T. (1989). *Multivariate Calibration.* Wiley.

Mejdell, T. (1990). *Estimators for Product Composition in Distillation Columns.* PhD thesis, University of Trondheim, The Norwegian Institute of Technology.

Moler, C., Little, J., Bangert, S., and Kleinman, S. (1990). *MATLAB User's Guide.* The MathWorks.

Ricker, N. L. (1988). The use of biased least-squares estimators for parameters in discrete-time pulse response models. *Industrial and Engineering Chemical Research*, 27:343–350.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society*, B.36:111–147.

Stone, M. and Brooks, R. J. (1990). Continuum regression: Cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares, and principal components regression. *Journal of the Royal Statistical Society*, B.52:237–269.

Wold, S., Ruhe, A., Wold, H., and Dunn, W. (1984). The collinearity problem in linear regression: The partial least squares approach to generalized inverses. *SIAM J. Sci. Stat. Comput.*, 5(3):753–743.

A Nomenclature

In general, script letters represent Hilbert spaces, capital letters represent matrices, lower case letters represent column vectors, and Greek letters represent scalars. Estimates are denoted by a tilde, "~". The dimensions of matrices are denote by subscripted n's.

some matrices, vectors and scalars

$\tilde{b}$	$n_i \times 1$	is the biased estimate of r. See equation 26 .
e	$n_s \times 1$	is the measurement noise corrupting output. See equation 1.
I	as appropriate	is the identity matrix.
M	$n_s \times n_s$	is matrix of weights for the weighted least squares objective function. See equation 7
P	$n_s \times n_s$	is the output error covariance matrix.
r	$n_i \times 1$	is the "true" regression vector. See equation 1.
$\tilde{r}$	$n_i \times 1$	the Ordinary Least Squares (OLS) regressor. See equation 2.
v	varies$\times 1$	is a vector locally defined. Any given v may or may not relate to any other v.
W	$n_i \times n_w$	is the matrix whose range defines the search space for $\tilde{b}$. See equation 26.
X	$n_s \times n_i$	is the input data; each row corresponds to one input sample. Thus, $X^T = [x_1 \;\; x_2 \;\; \ldots \;\; x_{n_s}]$.
x_j	$n_i \times 1$	is the jth input data sample.
y	$n_s \times 1$	is the measured output data for scalar output problems. See equation 1.
$\tilde{y}$	$n_s \times 1$	is the regression prediction for scalar output data.
y_i	$n_o \times 1$	is the ith output data sample for vector output problems.
z_i	$n_s \times 1$	is the vector produced by the ith input for all samples. Thus, $X = [z_1 \;\; \ldots \;\; z_{n_i}]$.
ψ_j	scalar	is the jth component of y. $y = [\psi_1, \ldots, \psi_{n_s}]$.
$\tau(w, y)$	scalar	is the test statistic for w and a given y.

dimensional descriptors

n_d is the number of "significant subspaces" to be generated.

n_i is the number of inputs.

n_s is the number of samples.

n_p is dimension of the allowable space in which to search for further w_i^{opt}. For scalar output problems, $n_p = n_i - i + 1$.

n_w is the rank of W.

operators

$|\cdot|$ is the absolute value.

$\|\cdot\|$ is the Euclidean norm. $a = \sqrt{< a, a >}$.

$[W\ |V]$ is the matrix formed by placing W and V side-by-side.

$< \cdot, \cdot >$ is the inner product. For matrices A and B, $< A, B > = \mathrm{Tr}(AB^T)$.

$\mathcal{E}(\cdot)$ is the expectation.

Range$(\cdot)$ is the range of an operator. For a matrix, the range is the span of the column vectors.

Tr$(\cdot)$ is the trace, the sum of the diagonal elements of a matrix.

Var$(\cdot)$ is the variance.

B Generation of Data for Simulation Examples

The simulation exploration was conducted using Matlab [Moler et al., 1990]. The two Matlab M-files used to generate the data are described below. The vector output examples were generated using the `gen_dat2_rob` routine with the parameters: `n_train = 30`, `n_test = 100`, `d = 10`, `o = 4`, `d_ind = 3`, `max_exp = 5`, `min_exp = 0`, `noise = 0.5`, and `e = 0.05`, `outvar = 3`. The results in Table 4 used `e = 0`, while the results in Table 5 used `outvar = 30`. The generation routine is specifically designed to produce difficult examples. The "true" regression vectors (columns of R) are drawn from a spherically

symmetric distribution about the origin (all directions are equally probable). However, the length of these vectors varies over 5 orders of magnitude. Thus, from a Bayesian viewpoint, the prior distribution for the regression vector is not particularly informative. The X are chosen independently of the R and the singular values (the square roots of the eigenvalues of X^TX) also vary over 5 orders of magnitude. Thus, there will be large variances in the X data which do not lie in any of the directions of the columns of R and therefore have little effect on the output. This will trouble principal component regression methods that proceed by examining directions in the order of the value of their singular values (principal components). Lastly, three of the explanatory variables vary independently of all other explanatory variables, but the remaining seven are correlated. This covariance structure can cause difficulties for both variable subset selection methods such as step-wise regression [Frank and Friedman, 1992] and for scaling methods such as auto-scaling (using "standardized variables") that weight the explanatory data solely on the variance of each individual explanatory variable.

B.1 Routine to generate random problems

```
function [X,y,Xt,yt,b] =gen_dat2_rob(n_train,n_test,d,o,d_ind,max_exp,
                                      min_exp,noise,e,out_var)

% this function generates data for linear regression problems
%
%
%  n_train  is the number of samples in the training set
%  n_test   is the number of samples in the testing set
%  d        is the number of inputs
%  o        is the number of outputs
%  d_ind    is the number of inputs NOT rotated and thus "independent"
%  max_exp  the largest order of magnitude contemplated
%  min_exp  the smallest order of magnitude contemplated
%            used for scaling the input data and
%            generating the regression vector
%  noise    std deviation of the normal additive noise
```

```
%
%
%  X          is the input training data
%  Xt         is the input testing data
%  y          is the output (noise corrupted) training data
%  yt          is the output (not noise corrupted) testing data



scale = diag(abs(scaled_rand(max_exp,min_exp,d)));
% these b's are for the same direction as singular vectors
for i=1:o
    b(:,i) = scaled_rand(max_exp,min_exp,d);

end

% need to build random orthogonal matrix
% only rotate d - d_ind columns; let the rest be
% 'approx' independent


d_rot =  d - d_ind;
if  d_ind == d
  v = eye(d);
else
  rand('uniform')
  v = rand (d_rot);
  [u,s,v] = svd(v);
  if d_rot == d
      v = u*v;
  else
      v = [ eye(d_ind), zeros(d_ind,d_rot); zeros(d_rot,d_ind), u*v];
```

```
   end

end


% use v as an additional rotation on the data and regression vector

rand('normal')
X =  rand(n_train,d) * scale * v;
Xt =  rand(n_test,d) * scale * v;
b = v'*b;
yt = Xt*b;
%desire RMS of null predictor to be 10
rms = sqrt(trace(yt'*yt)/(n_test* o) );
b=b/rms;
yt = Xt*b;
y = X*b;
% need to produce outliers
for i= 1:n_train
   for j = 1:o
      rand('uniform')
      if (e < rand)
        rand('normal')
        y(i,j) = y(i,j) + rand*noise;
      else
        rand('normal')
        y(i,j) = y(i,j) + rand*noise*out_var;
      end
   end
end
```

B.2 Routine to generate "exponential" random numbers

```
function vect =  scaled_rand(u,l,d)

% this function generates a vector of random numbers that are
% 'exponentially' distributed;  that is, the probability of
%  a number having any given  order of magnitude within
%  the valid range is roughly equal
%
%   u      lowest order of magnitude allowed
%   l      highest order of magnitude allowed
%   d      is the dimension of the vector generated
%
%  10^l < number < 10^u
%

rand('uniform');

for i = 1:d
  vect(i) = 10^ ( (u - l) * rand(1,1)  + l);
end

vect = vect';
```